

周报

1. 前述工作回顾

作为本周工作的基础，同时为方便阅读本周工作内容，因此先简单回顾下之前的工作。

在实现了相似哈希生成指纹的代码并进行了测试后，将之重构封装成 `SimHash` 类，以支持对体数据逐块进行相似哈希指纹计算，主要功能即是对输入（当前支持类型为 `uchar`）的串，按需进行特征定义，分割，权重计算，最终计算其相似哈希指纹。同时，汉明距离计算和相似度计算也包含在该类内。

上周对于指纹集合的聚类算法进行了一些实验和探索，由于指纹本身的数据属性（01 二进制串，不包含原始数据信息等特点）先后否定了 `google` 那篇文章所采用的相似查询算法和直接 `kmeans` 聚类方法。分析学习了谱聚类方法后，决定采用谱聚类方法作为相似哈希指纹的聚类算法进行试验。

当前的压缩端流程和最初设想的流程大致相同，如图 1，图 1 左图为旧的压缩端流程，右端为当前的压缩流程。

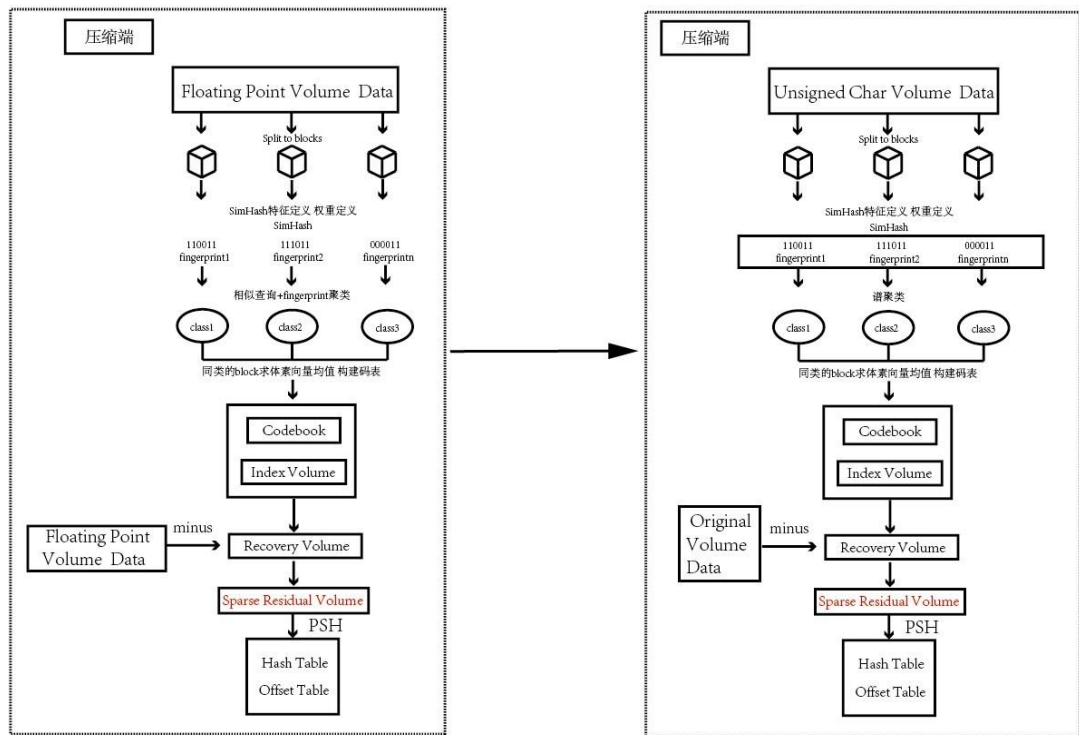


图 1 压缩端流程

当前主要改动有两处，1 是原来针对的是浮点型体数据，现在暂时先改为处理 unsigned char 类型的体数据；2 是对生成的体数据指纹集合明确地采用谱聚类方法进行分类以构建码表 codebook

2. 实现细节

为了实现图 1 的当前流程，本周在前述工作的基础上，首先设计实现一个简单的体数据分块处理模块 **VolumeSplitter**，主要负责体数据的加载、分块读取、处理等功能。将体数据分块加载、处理完成以后，逐块地将每块体数据值取出并转换成 string 表示，然后逐块地使用 **SimHash** 类计算其相似哈希指纹，得到体数据相似哈希指纹集合 **Q**。接着，实现一个名为 **SimHashCluster** 的类，该类主要负责相似哈希指纹集合的聚类并生成码表和索引体纹理。得到 **Q** 后，初始化 **SimHashCluster** 类实例，假设 **Q** 大小为 **N**，计算

Q 的相似矩阵(权重邻接矩阵) $W(N*N)$ 和拉普拉斯矩阵 $L(N*N)$ ，然后计算 L 的特征值和特征向量，特征值按增序排列，特征向量同样对应；取前 K 个特征向量进行 Kmeans 聚类，得到 C 个聚类结果；分类完毕。

之前，每个类和模块均已经独立测试过确保正确。在实现完上述功能后，再次对整个流程进行测试，同时观察该算法的基本运行结果，试验分类是否良好，哪些参数需要重点调试等。为此，构造了一个测试体数据，该数据大小为 $8*8*8$ ，共含 512 个体素，类型为 unsigned char，分块大小为 $4*4*4$ ，因此可分为 8 个块。测试分析见第三部分

3. 测试分析和反馈调节

测试实例一

将测试数据分成 8 个块，每一块的体数据值都设为同一个值，分别为 15, 47, 79, 111, 143, 175, 207, 239 间隔 32.对测试数据进行简单的体绘制，结果如图 2

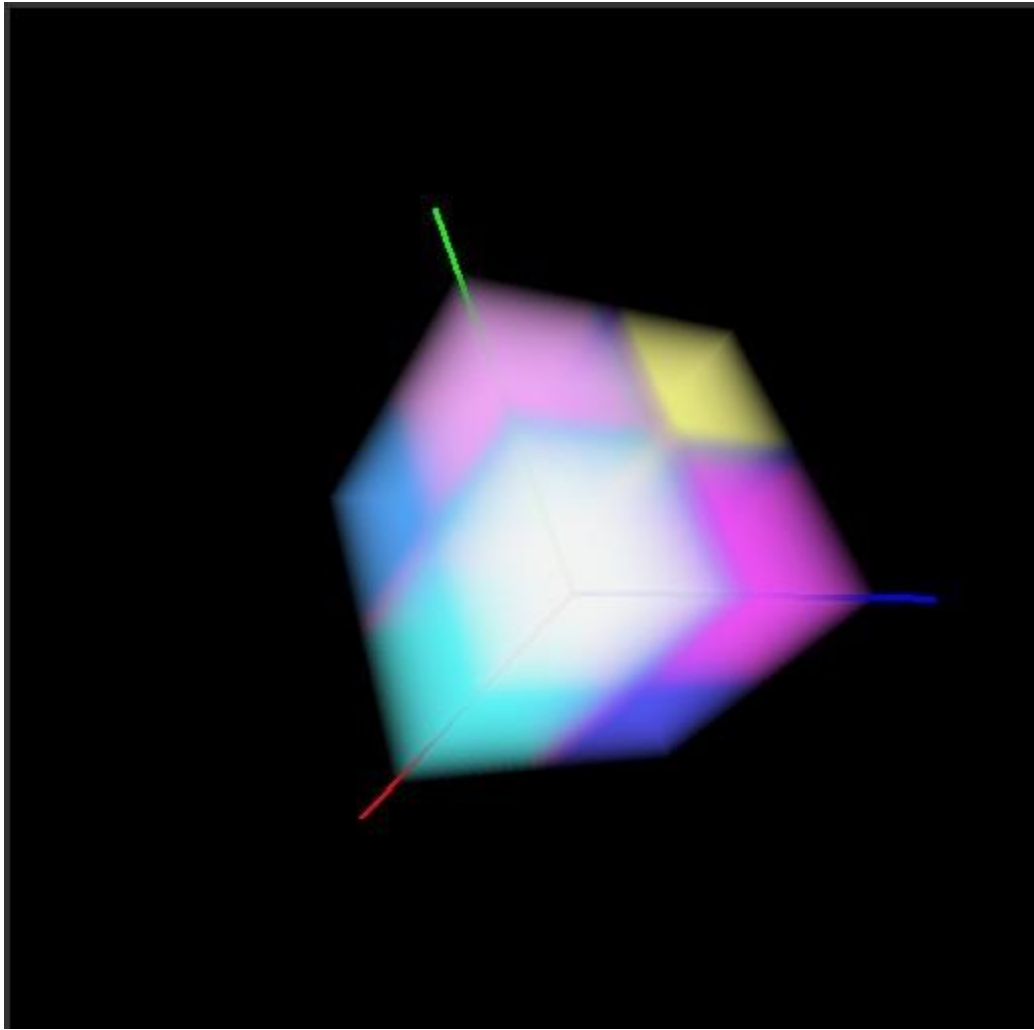


图 2 测试数据实例一直接图绘制结果

假设 8 个体数据块的编号为 0-7，我们可以得到如表格 1 所示的一张汉明距离图，表示每个体数据块所对应的相似哈希指纹之间的汉明距离，值越大表示越不相似。

	0	1	2	3	4	5	6	7
0	0	60	67	64	61	52	55	64
1	60	0	63	60	71	64	57	60
2	67	63	0	59	64	61	60	57

3	64	60	59	0	57	64	57	72
4	61	71	64	57	0	67	66	57
5	52	64	61	64	67	0	63	56
6	55	57	60	57	66	63	0	65
7	64	60	57	72	57	56	65	0

表 1 体数据块相似哈希指纹汉明距离图

定义两个相似哈希指纹 i 和 j 的相似性如公式 1

$$\text{Likeness} = (128 - \text{HammingDistance})/128$$

(1)

由此得到指纹集合的相似性矩阵 W ，如图 3

1	0.53125	0.476563	0.5	0.523438	0.59375	0.570313	0.5
0.53125	1	0.507813	0.53125	0.445313	0.5	0.554688	0.53125
0.476563	0.507813	1	0.539063	0.5	0.523438	0.53125	0.554688
0.5	0.53125	0.539063	1	0.554688	0.5	0.554688	0.4375
0.523438	0.445313	0.5	0.554688	1	0.476563	0.484375	0.554688
0.59375	0.5	0.523438	0.5	0.476563	1	0.507813	0.5625
0.570313	0.554688	0.53125	0.554688	0.484375	0.507813	1	0.492188
0.5	0.53125	0.554688	0.4375	0.554688	0.5625	0.492188	1

图 3 指纹集合的相似性矩阵

再构造对应的拉普拉斯矩阵 L ，如图 4 所示

3.69531	-0.53125	-0.476563	-0.5	-0.523438	-0.59375	-0.570313	-0.5
-0.53125	3.60156	-0.507813	-0.53125	-0.445313	-0.5	-0.554688	-0.53125
-0.476563	-0.507813	3.63281	-0.539063	-0.5	-0.523438	-0.53125	-0.554688
-0.5	-0.53125	-0.539063	3.61719	-0.554688	-0.5	-0.554688	-0.4375
-0.523438	-0.445313	-0.5	-0.554688	3.53906	-0.476563	-0.484375	-0.554688
-0.59375	-0.5	-0.523438	-0.5	-0.476563	3.66406	-0.507813	-0.5625
-0.570313	-0.554688	-0.53125	-0.554688	-0.484375	-0.507813	3.69531	-0.492188
-0.5	-0.53125	-0.554688	-0.4375	-0.554688	-0.5625	-0.492188	3.63281

图 4 对应 w 的拉普拉斯矩阵

计算 L 的特征值，并按升序排列，如图 5

```
-6.28598e-017
3.98878
4.03326
4.1022
4.13566
4.22043
4.26883
4.32896
```

图 5 L 的特征向值

其中第一行的特征值实际上应该是 0，由于计算机浮点误差所以此处显示不为 0。

依据谱聚类对于拉普拉斯矩阵属性的说明（参考文献 1）以及本测试数据的数据分布，可知图 5 所示的特征值实际表明聚类聚成一类是合乎特征值计算结果的（由参考文献 1 可知）。然而，实际上分成一类明显是不对的。或者说不合理的。原因见[分析](#)。

对应的特征向量如图 6 所示，

```
0.353553 0.122422 0.0613997 0.570809 -0.105209 -0.235906 -0.265871 0.
.626907
0.353553 0.486524 -0.104957 -0.111907 0.682047 0.189523 -0.311623 -0.
.128606
0.353553 0.0163808 0.0611042 -0.676865 -0.412848 -0.236292 -0.426451 0.
0686607
0.353553 -0.0699128 -0.661745 -0.0997848 -0.190605 0.471902 0.317305 0.
.250087
0.353553 -0.811112 -0.0821109 0.135561 0.27841 -0.0803127 -0.228093 -0.
.236619
0.353553 0.145136 0.345622 0.322818 -0.43047 0.445265 -0.0732379 -0.
.491264
0.353553 0.22568 -0.224936 0.104434 -0.0488495 -0.645624 0.448016 -0.
.377664
0.353553 -0.115117 0.605624 -0.245065 0.227523 0.0914453 0.539955 0.
.288498
```

图 6L 的特征向量

取前 8 个特征值所对应的特征向量聚成 8 类，结果如下

Cluster 0: Item 0

Cluster 1: Item 4

Cluster 2: Item 1

Cluster 3: Item 6

Cluster 4: Item 3

Cluster 5: Item 2

Cluster 6: Item 7

Cluster 7: Item 5

取前 8 个特征值所对应的特征向量聚成 4 类，结果如下

Cluster 0: Item 0 5 7

Cluster 1: Item 4

Cluster 2: Item 2 1 3

Cluster 3: Item 6

取前 8 个特征值所对应的特征向量聚成 2 类，结果如下

Cluster 0: Item 2 1 0 5 3 6 7

Cluster 1: Item 4

取前 4 个特征值所对应的特征向量聚成 8 类，结果如下

Cluster 0: Item 0

Cluster 1: Item 4

Cluster 2: Item 1

Cluster 3: Item 6

Cluster 4: Item 3

Cluster 5: Item 2

Cluster 6: Item 7

Cluster 7: Item 5

取前 4 个特征值所对应的特征向量聚成 4 类，结果如下

Cluster 0: Item 0 5

Cluster 1: Item 4

Cluster 2: Item 2 7 1

Cluster 3: Item 3 6

取前 4 个特征值所对应的特征向量聚成 2 类，结果如下

Cluster 0: Item 5 6 0 1 7

Cluster 1: Item 3 2 4

分析

为什么会出现上述蓝色字体所说的情况呢？参考图 3，该矩阵中的相似性即可认为是谱聚类过程中的边权重，由于谱聚类的分类主要是基于图的边权重，即同一类内边权重之和尽量大，不同类之间边权重尽量小这样的规则来分类，而图 3 所展示的，亦即我们的依据汉明距离

的相似度，若直接采用该相似度作为体数据块之间的相似度，极有可能无法很好的进行谱聚类（因为，两两相似度差异不够显著，实际的体数据值则差异较大）。因此，我对相似度进行了改进。

原先，我们直接采用两个体数据块所对应的指纹的汉明距离按公式 1 进行相似度构建。由于差异显著性不够谱聚类方法的内在要求，因此做了如下改进：

首先设定一个汉明距离相似性判断阈值 `_hamming_distance_threshold`，若两个体数据块所对应的指纹的汉明距离大于该阈值，直接设相似度为 0；否则，按公式 1 构建两者相似度。我们设阈值为 16，由于指纹总长度为 128，所以当汉明距离为 16 时，相似度为 87.5%。

由此，可得当前体数据块相似哈希指纹汉明距离图和表 1 是一样的，但是矩阵 **W** 和 **L**（对应图 3 和图 4）都发生了变化，**W** 变为一个 8*8 的单位阵，而 **L** 则是 8*8 的全 0 矩阵。接着，可计算 **L** 的特征值为 0，且是重根，重根度为 8，亦即 **L** 有 8 个为 0 的特征值。依据参考文献 1，可知此时应该聚成 8 个类比较适合，而实际的计算结果也支持这一说法，由此可见我们的改进方法是可行的。下面是分情况的聚类结果。

取前 8 个特征值所对应的特征向量聚成 8 类，结果如下

Cluster 0: Item 0

Cluster 1: Item 4

Cluster 2: Item 1

Cluster 3: Item 6

Cluster 4: Item 3

Cluster 5: Item 2

Cluster 6: Item 7

Cluster 7: Item 5

取前 8 个特征值所对应的特征向量聚成 4 类，结果如下

Cluster 0: Item 5 7 0 2 3

Cluster 1: Item 4

Cluster 2: Item 1

Cluster 3: Item 6

取前 8 个特征值所对应的特征向量聚成 2 类，结果如下

Cluster 0: Item 3 5 6 7 0 1 2

Cluster 1: Item 4

取前 4 个特征值所对应的特征向量聚成 8 类，结果如下

Cluster 0: Item 0

Cluster 1: Item 4 5 6 7

Cluster 2: Item 1

Cluster 3: Item

Cluster 4: Item 3

Cluster 5: Item 2

Cluster 6: Item

Cluster 7: Item

取前 4 个特征值所对应的特征向量聚成 4 类，结果如下

Cluster 0: Item 0

Cluster 1: Item 2 3

Cluster 2: Item 1

Cluster 3: Item 4 5 6 7

取前 4 个特征值所对应的特征向量聚成 2 类，结果如下

Cluster 0: Item 0

Cluster 1: Item 4 5 6 7 1 2 3

经过上述测试和验证，可知几个重要的调节参数：

1. 汉明距离相似性判断阈值_hamming_distance_threshold

2. 聚类特征向量个数 K

3. 聚类个数 C

文献 1 中有部分关于参数 2 和 3 的研究总结,可以借鉴,但是还不够,最终肯定需要反复调节测试。参数 1 和数据本身值的分布、特征定义、特征权重定义密切相关,需要再进行摸索

测试实例二

为了进一步验证方法是否可靠,制作了新的测试数据,将测试数据分成 8 个块,分块方式如测试实例 1,每一块的体数据值,分别为 15, 15, 79, 79, 143, 143, 207, 207.对测试数据进行简单的体绘制,传输函数和测试一一致,结果如图 7

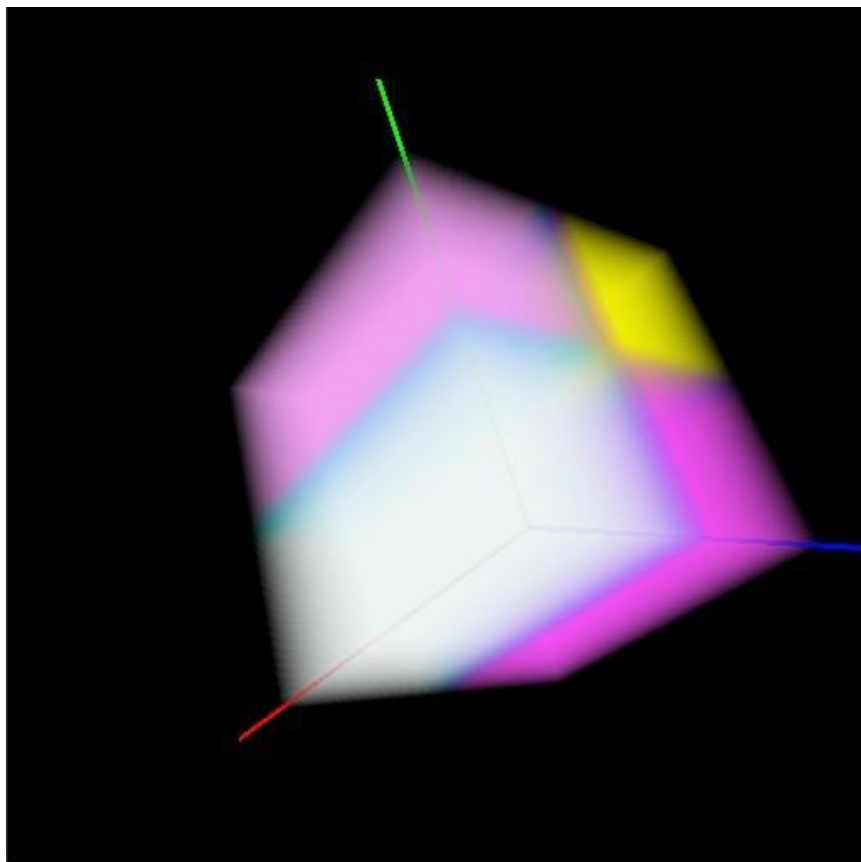


图 7 测试数据实例二直接图绘制结果

汉明距离图如表格 2。

	0	1	2	3	4	5	6	7
0	0	0	67	67	61	61	55	55
1	0	0	67	67	61	61	55	55
2	67	67	0	0	64	64	60	60
3	67	67	0	0	64	64	60	60
4	61	61	64	64	0	0	66	66
5	61	61	64	64	0	0	66	66
6	55	55	60	60	66	66	0	0
7	55	55	60	60	66	66	0	0

表 2 体数据块相似哈希指纹汉明距离图

首先直接采用两个体数据块所对应的指纹的汉明距离按公式 1 进行相似度构建，由此得到指纹集合的相似性矩阵 W，如图 8

1	1	0.476563	0.476563	0.523438	0.523438	0.570313	0.570313
1	1	0.476563	0.476563	0.523438	0.523438	0.570313	0.570313
0.476563	0.476563	1	1	0.5	0.5	0.53125	0.53125
0.476563	0.476563	1	1	0.5	0.5	0.53125	0.53125
0.523438	0.523438	0.5	0.5	1	1	0.484375	0.484375
0.523438	0.523438	0.5	0.5	1	1	0.484375	0.484375
0.570313	0.570313	0.53125	0.53125	0.484375	0.484375	1	1
0.570313	0.570313	0.53125	0.53125	0.484375	0.484375	1	1

图 8 指纹集合的相似性矩阵

再构造对应的拉普拉斯矩阵 L，如图 9 所示

4.14063	-1	-0.476563	-0.476563	-0.523438	-0.523438	-0.570313	-0.570313
-1	4.14063	-0.476563	-0.476563	-0.523438	-0.523438	-0.570313	-0.570313
-0.476563	-0.476563	4.01563	-1	-0.5	-0.5	-0.53125	-0.53125
-0.476563	-0.476563	-1	4.01563	-0.5	-0.5	-0.53125	-0.53125
-0.523438	-0.523438	-0.5	-0.5	4.01563	-1	-0.484375	-0.484375
-0.523438	-0.523438	-0.5	-0.5	-1	4.01563	-0.484375	-0.484375
-0.570313	-0.570313	-0.53125	-0.53125	-0.484375	-0.484375	4.17188	-1
-0.570313	-0.570313	-0.53125	-0.53125	-0.484375	-0.484375	-1	4.17188

图 9 对应 w 的拉普拉斯矩阵

计算 L 的特征值，并按升序排列，如图 10

```
-2.19295e-015
3.9863
4.03068
4.32677
5.01562
5.01563
5.14062
5.17187
```

图 10 L 的特征值

对应的特征向量如图 11 所示

```
      -0.353553      -0.199023      -0.361719      -0.452272      0
0      -0.707107      0
      -0.353553      -0.199023      -0.361719      -0.452272      -2.24372e-016      6.21379e-
016      0.707107      2.09378e-016
      -0.353553      0.523479      0.269011      -0.169125      -0.514682      -0.484
874      9.02436e-016      -5.02376e-015
      -0.353553      0.523479      0.269011      -0.169125      0.514682      0.484
874      5.25594e-016      -8.99281e-015
      -0.353553      -0.420805      0.428777      0.11863      0.484874      -0.514
682      1.02221e-015      6.07847e-015
      -0.353553      -0.420805      0.428777      0.11863      -0.484874      0.514
682      4.72333e-016      2.42584e-014
      -0.353553      0.0963491      -0.33607      0.502766      8.05595e-015      -2.43694e-
014      -9.58516e-016      0.707107
      -0.353553      0.0963491      -0.33607      0.502766      -7.32064e-015      -1.39333e-
014      -7.06575e-016      -0.707107
```

图 11 L 的特征向量

取前 8 个特征值所对应的特征向量聚成 8 类，结果如下

Cluster 0: Item 0

Cluster 1: Item 4

Cluster 2: Item 1

Cluster 3: Item 6

Cluster 4: Item 3

Cluster 5: Item 2

Cluster 6: Item 7

Cluster 7: Item 5

取前 8 个特征值所对应的特征向量聚成 4 类，结果如下

Cluster 0: Item 0

Cluster 1: Item 4 2 3

Cluster 2: Item 1

Cluster 3: Item 6 7 5

取前 8 个特征值所对应的特征向量聚成 2 类，结果如下

Cluster 0: Item 0

Cluster 1: Item 2 4 7 5 3 6 1

取前 4 个特征值所对应的特征向量聚成 8 类，结果如下

Cluster 0: Item 0

Cluster 1: Item 4

Cluster 2: Item 1

Cluster 3: Item 6

Cluster 4: Item 3

Cluster 5: Item 2

Cluster 6: Item 7

Cluster 7: Item 5

取前 4 个特征值所对应的特征向量聚成 4 类，结果如下

Cluster 0: Item 2 3

Cluster 1: Item 4 5

Cluster 2: Item 0 1

Cluster 3: Item 7 6

取前 4 个特征值所对应的特征向量聚成 2 类，结果如下

Cluster 0: Item 2 7 0 1 3 6

Cluster 1: Item 4 5

由图 10L 特征值的计算，只有一个特征值为 0 可知，分成一类是数学上比较符合谱聚类特性的分类方式，但是实际上，由数据分布和构成，我们可以知道分成 4 类，如“取前 4 个特征值所对应的特征向量聚成 4 类”这个分类结果是最好的。两者有矛盾，由此再次采用改进的相似度计算方法进行构建，则汉明距离还是如表格 2 所示，指纹集合的相似性矩阵 W，如图 12

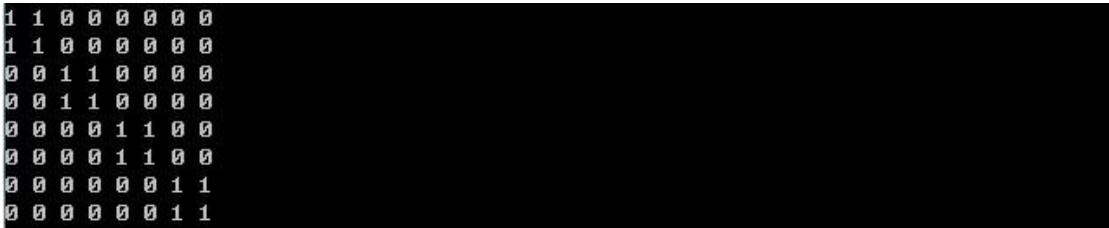


图 12 指纹集合的相似性矩阵

再构造对应的拉普拉斯矩阵 L ，如图 13 所示

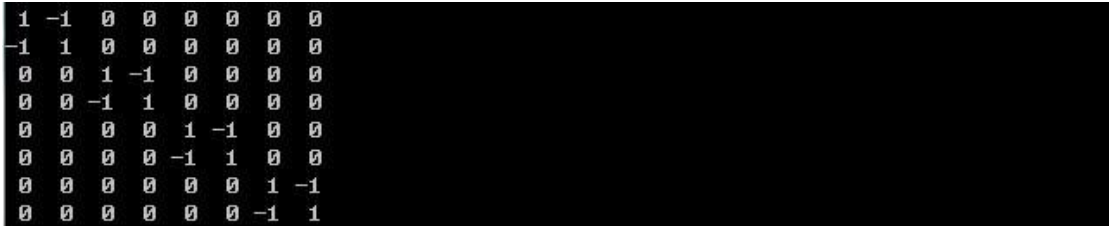


图 13 对应 w 的拉普拉斯矩阵

计算 L 的特征值，并按升序排列，如图 14

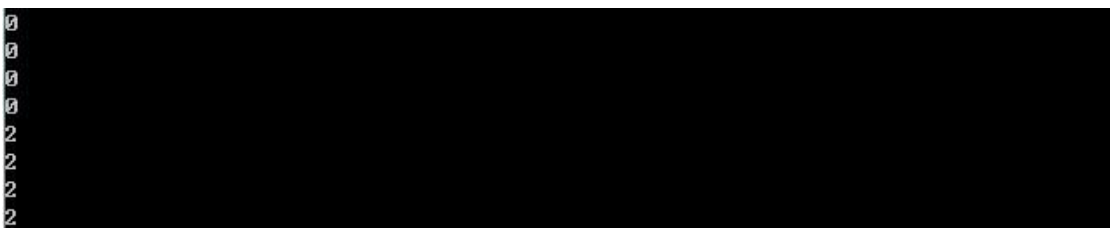


图 14 L 的特征值

对应的特征向量如图 15 所示

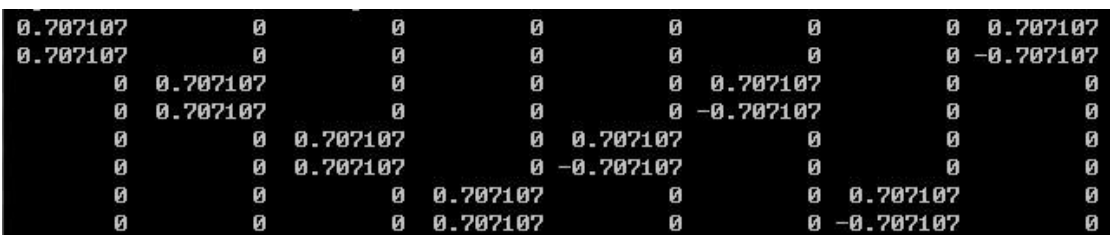


图 15 L 的特征向量

此时，由特征值可知，有四个值为 0 的特征值，因此，分为 4 类比较符合谱聚类的内在特性，同时，这也与实际的测试数据的数据值分布吻合。下面来看一下不同的分类结果。

取前 8 个特征值所对应的特征向量聚成 8 类，结果如下

Cluster 0: Item 0

Cluster 1: Item 4

Cluster 2: Item 1

Cluster 3: Item 6

Cluster 4: Item 3

Cluster 5: Item 2

Cluster 6: Item 7

Cluster 7: Item 5

取前 8 个特征值所对应的特征向量聚成 4 类，结果如下

Cluster 0: Item 2 3 0 5 7

Cluster 1: Item 4

Cluster 2: Item 1

Cluster 3: Item 6

取前 8 个特征值所对应的特征向量聚成 2 类，结果如下

Cluster 0: Item 0 1 2 3 5 6 7

Cluster 1: Item 4

取前 4 个特征值所对应的特征向量聚成 8 类，结果如下

Cluster 0: Item 0 1

Cluster 1: Item 4 5

Cluster 2: Item

Cluster 3: Item 6 7

Cluster 4: Item 2 3

Cluster 5: Item

Cluster 6: Item

Cluster 7: Item

取前 4 个特征值所对应的特征向量聚成 4 类，结果如下

Cluster 0: Item 2 3

Cluster 1: Item 4 5

Cluster 2: Item 0 1

Cluster 3: Item 6 7

取前 4 个特征值所对应的特征向量聚成 2 类，结果如下

Cluster 0: Item 0 1 2 3 6 7

Cluster 1: Item 4 5

4. 初步结论

通过对两个不同测试数据，分别用两种相似度计算方法构建谱聚

类相似性矩阵进行的测试结果可得出初步结论：

1. 聚类特征向量个数 K 和聚类个数 C 尽量相近 ($K \geq C$)，此时的聚类结果在同等条件下最好，最符合数据分布实际和谱聚类的内在特性
2. 使用改进的相似度计算方法，聚类结果明显要好于直接使用汉明距离进行相似度计算的对应结果，且符合数据分布和谱聚类内在要求
3. 上述两个测试数据实例中最好的结果分别用绿色字体予以高亮
4. 重要的参数调节中，只有汉明距离相似性判断阈值 `_hamming_distance_threshold` 目前还没有很好的方法予以设置，只能在实际应用中依据不同数据进行测试调校
5. 对于测试数据，由于事先已经知道数据分布特征，故可按上述方式进行测试，实际数据不知道数据分布，是否就无法进行类似的操作呢？答案是否定的，由于我们的测试都是以体数据值作为特征，出现频度作为特征值（权重），因此，使用体数据值的分布直方图，应该即可大概得出数据分布趋势，然后可以以其分布特征决定 C 和 K 。当然只有数据分布类别比较清晰的情况下，才可以很好的使用为 0 的特征值个数（或者比较稳定的前 N 个特征值）作为聚类个数，如果数据分布比较模糊，还是需要进一步设置。如图 16

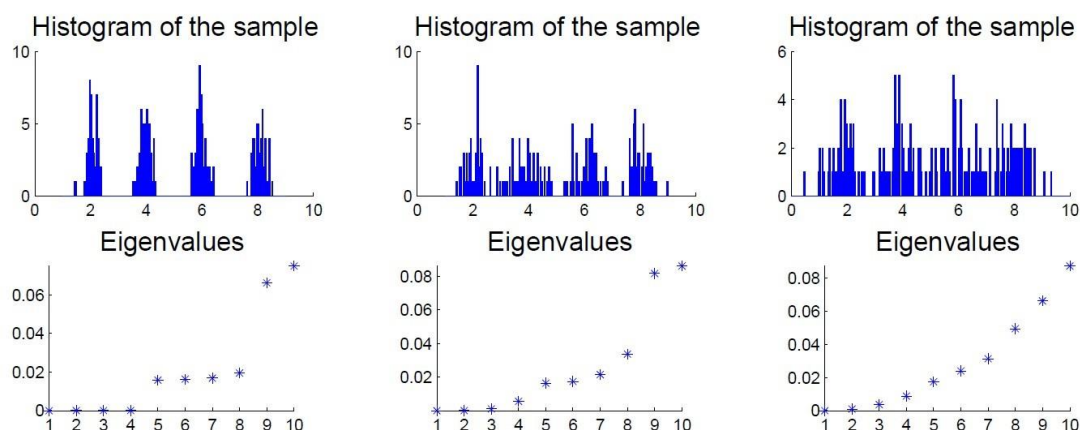


图 16 数据分布与特征值取值关系

图 16 第一行表示体数据值分布直方图，下一行表示对应的拉普拉斯矩阵特征值，总共 10 个特征值。我们可取数据值为 0 的特征值的个数或者，前 N 个稳定变化的特征值的个数为聚类个数。

5. 下周工作

至此，压缩端的核心算法已经实现并测试完毕！

1. 依据图 1 的压缩流程，聚类后，构建码表和索引体纹理；然后可以测试压缩率和压缩结果，如残差率等，压缩结果依赖于聚类结果；最后把这些数据送到 GPU 端进行解码绘制。
2. 针对不同体数据调试上述参数 1,2,3 对于结果的影响；还可考虑借鉴向量量化中关于体数据块大小的数学公式。
3. 针对真实数据进行测试验证

参考文献

1. **A tutorial on spectral clustering. Ulrike von Luxburg. Statistics and Computing, 17(4), 2007.**